OAUTH UND AUTORISIERUNGSMODULE IN SERVICE ORIENTED ARCHITECTURE

Am Beispiel eines sozialen Netzwerkes

PRODUKT

- Aktivitäts- und zeitmanagementorientiertes soziales Netzwerk
- eine Alternative zum klassischen Ticketsystem (Jira)
- Auf Privatpersonen orientiert
- Schwerpunkt: Datensicherheit und Datenschutz

Rechtliche Anforderungen

- **TMG § 5 Allgemeine Informationspflichte:** „unmittelbar erreichbares“ Impressum
- **TMG § 13 Pflichten des Diensteanbieters:** Umfang und Zwecke der Erhebung und Verwendung personenbezogener Daten
- **UrhG § 22:** Gesetz betreffend das Urheberrecht an Werken der bildenden Künste und der Photographie

TMG, TKG usw. (Social Media Recht)

Architekturkonzept

- **Umstieg von monolithische Architektur zu SOA:** Security-Verfahren wird möglichst flexibel gestaltet, sodass es mehrmals für verschiedene Zwecke verwendet werden könnte.
- **Auslagern von Autorisierungslogik:** Security Module auf Java mit Spring Security. Dies bietet unterschiedliche Designmuster und Konzepte für ein sicheres und zuverlässiges Security-verfahren an. Das Verschlüsseln von Passwörtern erfolgt mit bcrypt Algorithmus.
- **Verwendung von Frameworks**
- **Hierarchische Berechtigungssystem:** Die Zugriffsrechte in unserem System sind in 8 Schichten aufgeteilt (0 = nicht angemeldete User, >5 = interne Mitarbeiter u.s.w.).

Web request

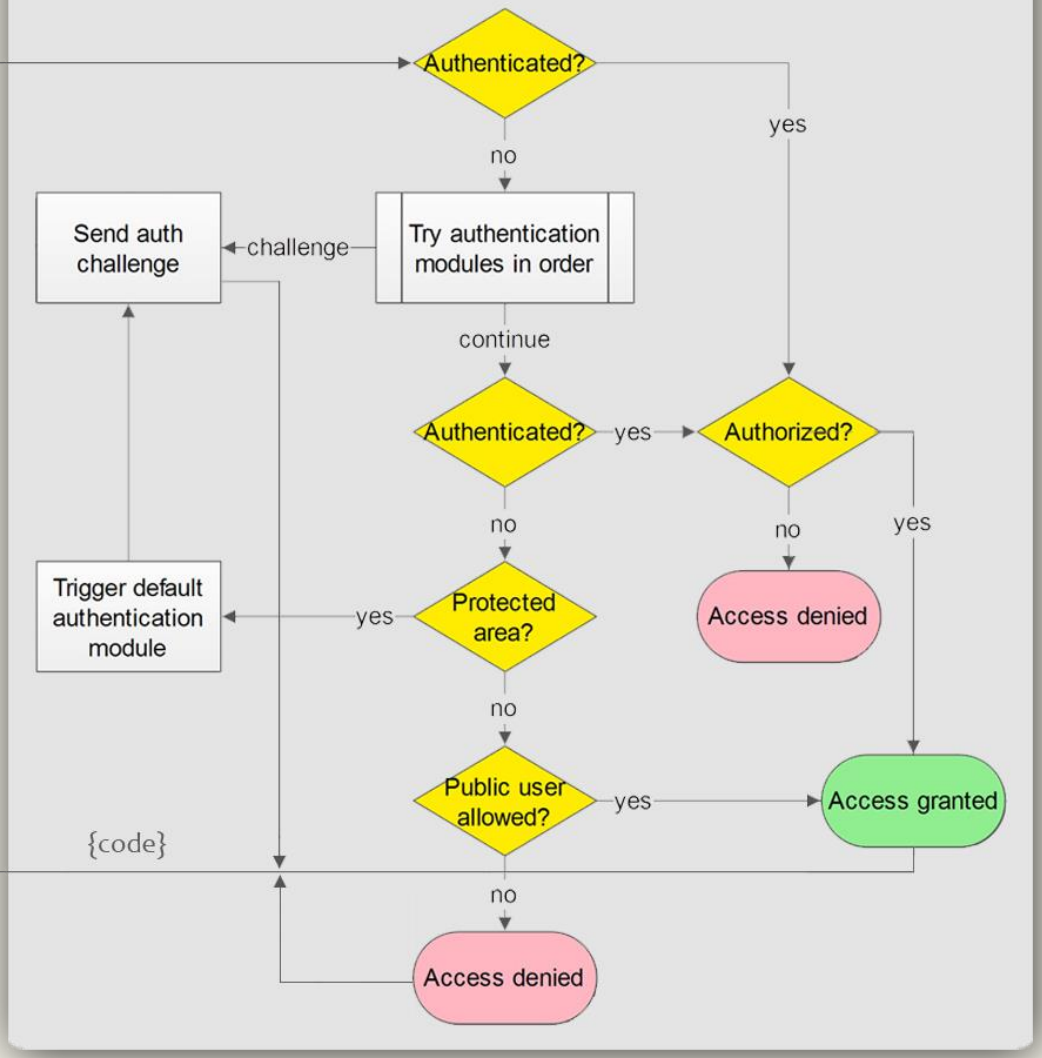
Web response

PRESENTATION LAYER
(Web, Mobile, Desktop)

{JSON}

REST API

Security module (presentation)



Security REST API

{JSON}

Security Core

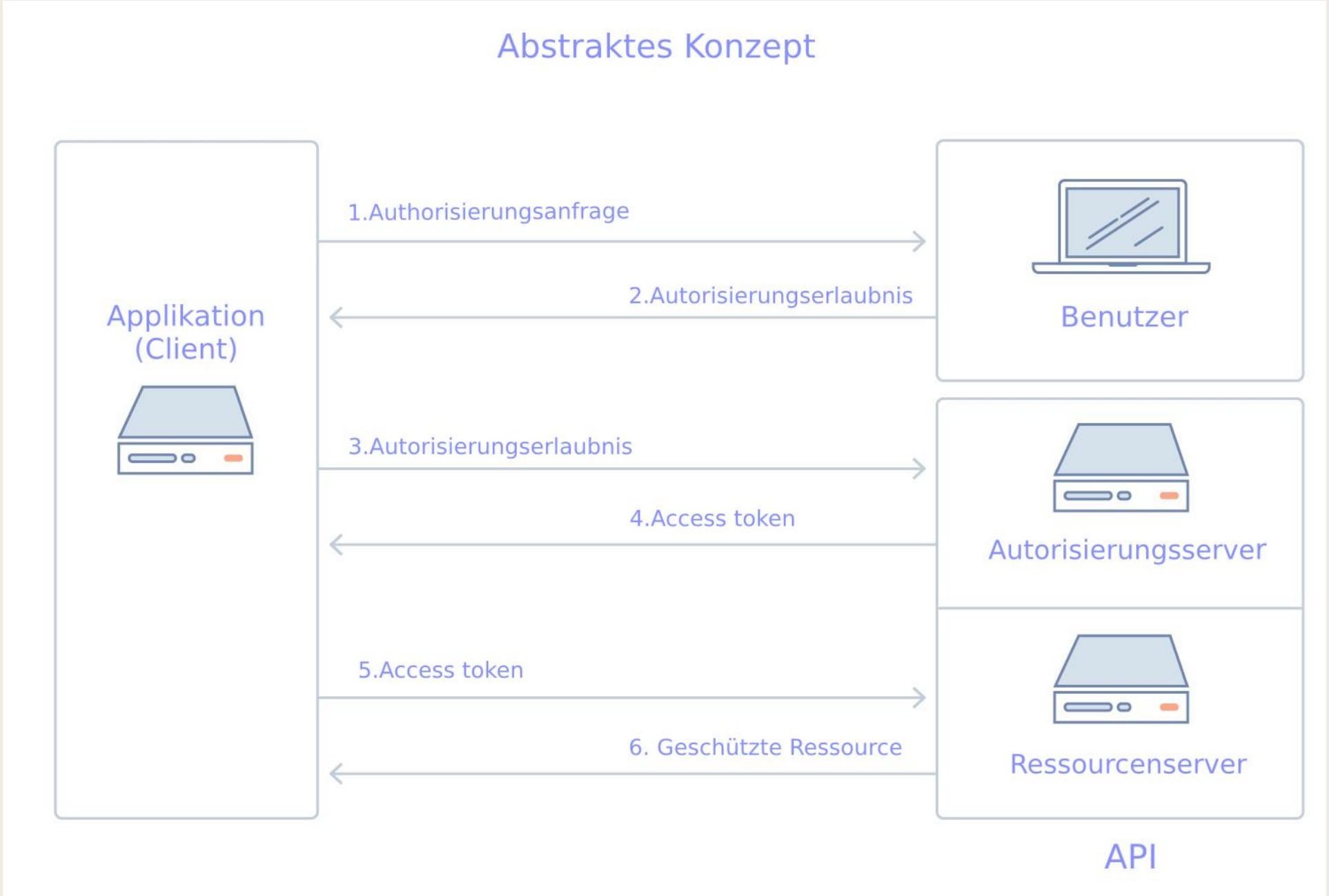
AuthDB

OAuth v2.0

- OAuth 2 = Autorisierungsframework
- Eingeschränkter Zugriff auf Benutzerkonten in HTTP-Diensten
- OAuth (Open Authorization) ist ein offenes Protokoll
- Unterstütz Web-, sowie Desktop- bzw. Mobile Applikationen.

Systemhierarchie und Konzept:

- **Kontoinhaber:** Der Benutzer autorisiert sein Konto. Der Zugriff einen Dienst auf dem Benutzerkonto ist beschränkt (lesende- oder schreibende Zugriff).
- **Ressourcenserver und Autorisierungsserver: API** Der Ressourcenserver speichert die geschützten Daten von Benutzerkonten und der Autorisierungsserver verifiziert die Authentizität der vom Benutzer bereitgestellten Informationen und erstellt dann Autorisierungstoken für die Anwendung, über die die Anwendung auf Benutzerdaten realisiert wird. Aus Sicht des Anwendungsentwicklers führt die Service-API sowohl die Ressourcenserverrolle als auch die Autorisierungsserverrolle aus.
- **Client: externer Service**
  Applikation bzw. Dienst, die auf das Benutzerkonto zugreifen, muss vor dem Zugriff vom Benutzer autorisiert werden und die Autorisierung muss von der API kontrolliert werden.


Abstraktes Konzept
Applikation
(Client)
1.Authorisierungsanfrage
2.Autorisierungserlaubnis
3.Autorisierungserlaubnis
4.Access token
5.Access token
6. Geschützte Ressource
Benutzer
Autorisierungsserver
Ressourcenserver
API